UNIVERSITÀ DEGLI STUDI DI PADOVA

UNIVERSITY OF PADOVA

DEPARTMENT OF MANAGEMENT ENGINEERING
DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

SUSTAINABLE MOBILITY

Supervisors:
MARTA DISEGNA

Master Candidate:
NIMA KARIMI

SUMMER 2023

Nima Karimi
Sustainable Mobility
Documentation, September 05, 2023
Supervisors: Marta Disegna

University of Padova
Dipartimento di Tecnica e Gestione dei sistemi industriali
Department of Management Engineering
Padova

Abstract

This thesis presents a comprehensive study on developing a machine learning model to predict mobility patterns in various weather conditions and group structures. The primary focus of this project is to assess the effectiveness of machine learning models in forecasting mobility behavior, considering different weather scenarios, social contexts, and route characteristics. Additionally, the research aims to identify the key factors influencing mobility behavior and explore the potential application of machine learning models in personalized mobility prediction and sustainable transportation planning.

To achieve these objectives, a range of machine learning models will be evaluated for their accuracy and precision in forecasting mobility patterns. The research will investigate how different models perform in predicting mobility behavior under varying weather conditions and social contexts. Moreover, an in-depth analysis will be conducted to identify the factors that significantly impact the accuracy of these models.

By examining the accuracy and precision of various machine learning models, this research will provide valuable insights into their effectiveness in mobility forecasting. Furthermore, it will uncover the factors that play a crucial role in influencing the accuracy of these models. The findings of this study will contribute to the advancement of machine learning applications in transportation planning and assist in developing personalized mobility prediction systems for sustainable transportation.

Keywords: machine learning, mobility patterns, weather conditions, group structures, accuracy, precision, personalized mobility prediction, sustainable transportation planning

Astratto

Questa tesi presenta uno studio approfondito per lo sviluppo di un modello di apprendimento automatico per prevedere i modelli di mobilità in diverse condizioni meteorologiche e strutture di gruppo. Il focus principale di questo progetto è valutare l'efficacia dei modelli di apprendimento automatico nella previsione del comportamento di mobilità, considerando diversi scenari meteorologici, contesti sociali e caratteristiche del percorso. Inoltre, la ricerca mira a identificare i fattori che influenzano il comportamento di mobilità ed esplorare la possibilità di utilizzare i modelli di apprendimento automatico per la previsione della mobilità personalizzata e la pianificazione del trasporto sostenibile.

Per raggiungere questi obiettivi, verranno valutati diversi modelli di apprendimento automatico per la loro accuratezza e precisione nella previsione dei modelli di mobilità. La ricerca indagherà su come diversi modelli si comportano nella previsione del comportamento di mobilità in diverse condizioni meteorologiche e contesti sociali. Inoltre, verrà condotta un'analisi approfondita per identificare i fattori che influenzano in modo significativo l'accuratezza di questi modelli.

Attraverso l'esame dell'accuratezza e della precisione dei vari modelli di apprendimento automatico, questa ricerca fornirà preziose intuizioni sulla loro efficacia nella previsione della mobilità. Inoltre, svelerà i fattori che giocano un ruolo cruciale nell'influenzare l'accuratezza di questi modelli. I risultati di questo studio contribuiranno all'avanzamento delle applicazioni di apprendimento automatico nella pianificazione dei trasporti e aiuteranno nello sviluppo di sistemi di previsione della mobilità personalizzata per il trasporto sostenibile.

Parole chiave: apprendimento automatico, modelli di mobilità, condizioni meteorologiche, strutture di gruppo, accuratezza, precisione, previsione della mobilità personalizzata, pianificazione del trasporto sostenibile.

Acknowledgement

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

1 Introduction

With the rapid expansion of cities and the escalating worries regarding the sustainable use of our natural resources, the quest for achieving transportation systems that are not only efficient but also eco-friendly has garnered significant attention and become an urgent global challenge. Sustainable mobility, often referred to as sustainable transportation, aims to develop and implement strategies that promote the efficient movement of people and goods while minimizing negative environmental and social impacts.

Furthermore, machine learning, a field of artificial intelligence, involves the development and application of computational models that can automatically learn and improve from data without explicit programming. By analyzing large and diverse datasets, machine learning algorithms can identify patterns, make predictions, and gain valuable insights that aid decision-making processes. In the context of sustainable mobility, machine learning can provide innovative solutions to optimize transportation systems, improve accessibility, and minimize environmental impact.

Optimizing transportation systems through machine learning involves the application of data analytics to enhance operational efficiency, reduce congestion, and improve the overall performance of transportation networks. By processing large volumes of data collected from sources such as traffic sensors, GPS devices, and social media platforms, machine learning algorithms can generate real-time traffic predictions, optimize routing, and dynamically adapt transportation services. These advancements lead to improved travel experiences, reduced travel times, and more efficient allocation of resources.

Accessibility is a key aspect of sustainable mobility, aiming to ensure that transportation services are available and equitable for all individuals, regardless of their physical abilities, income levels, or geographic location. Machine learning algorithms can help address accessibility challenges by analyzing data on travel demand, demographics, and infrastructure characteristics. This enables the identification of underserved areas, optimization of transit routes, and the design of transportation services that cater to the needs of diverse populations.

Environmental sustainability is another critical dimension of sustainable mobility. Transportation accounts for a significant portion of global greenhouse gas emissions and is a major contributor to air pollution. Machine learning can play a crucial role in minimizing the environmental impact of transportation by developing predictive models that estimate emissions, optimize energy consumption, and facilitate the integration of clean and renewable energy sources. Additionally, machine learning techniques can aid in the design of eco-routing algorithms, which suggest the most environmentally friendly travel routes based on real-time data.

In summary, the combination of machine learning and sustainable mobility presents a promising framework to address the complex challenges faced by transportation systems. By harnessing the power of machine learning algorithms, transportation stakeholders can optimize operations, improve accessibility, and mitigate environmental impact. As the demand for sustainable and efficient mobility solutions continues to grow, machine learning can provide valuable insights and tools to shape the future of transportation, leading to more sustainable, accessible, and environmentally friendly mobility systems.

1.1 Machine Learning

Can computers learn like humans do from experience? Yes, through machine learning (ML). It enhances system performance by using computational methods to learn from experience, mainly in the form of data. The key task in machine learning is developing algorithms that build models from data. By inputting experience data into these algorithms, we obtain models capable of predicting outcomes for new observations. In the context of computer science, machine learning focuses on learning algorithms [14].

[9] provides a more formal definition: “A computer program is said to learn from experience E for some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.“ The term "model" is broadly used to denote outcomes learned from data. In some literature (e.g. [4]), "model" might mean a global outcome, while "pattern" signifies a local outcome.

The procedure of constructing models from data using machine learning algorithms is referred to as learning or training. The data employed during this phase is known as training data, where each sample serves as a training example, and the complete collection of training examples is the training set. The model derived from this process, representing the underlying data rules, is also termed a hypothesis, while the actual underlying rules are considered as facts or ground-truth. Thus, the fundamental goal of machine learning is to identify or approximate the ground-truth rules [14].

A wide range of machine-learning algorithms has been developed to address the diverse data and problem types in various machine-learning scenarios [5, 10]. These algorithms can be understood as exploring a vast space of potential programs, guided by training experience, to discover the most optimal performance. Variation among these algorithms arises from their methods of representing candidate programs, such as decision trees or mathematical functions, and their strategies for navigating this program space, including optimization techniques and evolutionary search methods [7].

1.1.1 Different Machine Learning algorithms

Machine learning encompasses a wide range of learning types and algorithms. The major categories are supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [1].

Supervised learning algorithms are trained on labeled data, which includes inputs and desired outputs. Common supervised tasks are classification, predicting categorical outputs like spam or not spam, and regression, predicting continuous values like price. Algorithms learn a mapping from inputs to outputs by examining many examples. Supervised learning can achieve high predictive accuracy but relies on large labeled training sets which can be costly to obtain. Popular supervised algorithms include logistic regression, support vector machines, neural networks, decision trees, and random forests.

In contrast, unsupervised learning operates on unlabeled data, finding hidden patterns and intrinsic structure within it. Clustering algorithms are a key example, grouping data points that are similar. Other unsupervised techniques like dimensionality reduction and association rule mining also derive insights from the data itself without external labeling. As labeling is not required, unsupervised methods can more easily scale to new problem domains, but the models may not have clear accuracy measures.

In summary, machine learning employs various approaches to train models that can analyze data, recognize patterns, make predictions, or perform tasks. Supervised learning offers predictive accuracy but requires labeled data. Unsupervised learning can find hidden insights in any data without labeling but has less defined performance measures. Together these technologies enable machine learning systems to perform tasks not explicitly programmed.

1.1.2 Overfitting

In supervised machine learning, an issue called overfitting arises, leading to poor generalization from observed data to new, unseen data. This results in a model performing well on the training set but poorly on the testing set. Overfitting occurs because the model becomes too specific to the training data and struggles with variations present in the testing data. Overfitted models tend to memorize noise and details of the training set rather than learning the underlying patterns [13].

Occam's Razor, or the principle of parsimony, advocates using models that contain only what's necessary for effective modeling. Overfitting occurs when models or procedures include more terms than needed, violating parsimony. Two types of overfitting are identified: using excessively flexible models and incorporating irrelevant components. Overfitting is undesirable due to resource wastage, potential prediction errors, worse decisions in feature selection, degraded predictions, and reduced portability of models. Portable models, adhering to Occam's Razor, are preferred for their broader applicability and reproducibility across locations [6, 3].

1.1.3 Spliting the dataset

In both statistical and machine learning model development, a common practice is to divide the dataset into two parts: training and testing [5]. The training set is utilized to estimate the model's unknown parameters, while the model's accuracy is assessed using the testing dataset. This division is essential to prevent overfitting, where a model becomes too tailored to the data, potentially leading to poor predictions in new situations. By reserving a subset of the data for testing, the model's performance can be evaluated before actual deployment, safeguarding against issues arising from overfitting.

The simplest and probably the most common strategy to split such a dataset is to randomly sample a fraction of the dataset. For example, 80% of the rows of the dataset can be randomly chosen for training, and the remaining 20% can be used for testing. Various strategies are showcased in [8] for efficiently and optimally dividing the dataset.

1.1.4 Validation

Furthermore, it's a frequent practice to reserve a segment of the training set for validation. This validation subset serves purposes such as refining model performance through hyper-parameter or regularization parameter selection (e.g. the number of hidden units—layers and layer widths—in a neural network [12]). Validation datasets also can be used for regularization by early stopping (stopping training when the error on the validation data set increases, as this is a sign of over-fitting to the training data set) [11].

To ensure greater result stability and maximize the use of valuable data for training, datasets can be repeatedly divided into multiple training and validation subsets. This practice, referred to as cross-validation, is employed. Additionally, an independent test dataset held apart from cross-validation is typically utilized to validate the model's performance. [2] provides an overview of various cross-validation techniques.

1.2 Thesis Structure

Chapter 2

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Chapter 3

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Chapter 4

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Chapter 4

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Chapter 5

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

2 Data Analysis

> *Information is the oil of the 21st century, and analytics is the combustion engine.*
>
> — **Peter Sondergaard**
> (Founder of The Sondergaard Group, LLC.)

This chapter discusses data analysis methodologies. Collaborating with Mowi Space, a platform tailored for mountain biking enthusiasts and winter sports enthusiasts, this collaboration has yielded a dynamic digital platform catering to outdoor sports. The MOWI website offers real-time data and interactive 3D maps for offline exploration. It delivers current trail conditions, weather updates, lift operations, and local events, ensuring informed and safe adventures. Notably, the Live Track feature allows real-time monitoring of family or friends during mountain activities, fostering connectivity and safety. This collaboration signifies a pivotal advancement in outdoor experience planning, merging technology with nature"s allure.

As a result of getting some user"s data going through various tracks, it"s possible to convert the GPS data into CSV files.

2.1 Converting the GPS data

This Python code snippet performs data processing on GPS data from GPX files and converts it into a more analyzable CSV format. The code utilizes several libraries for various functionalities.

2.1.1 Imports

The script begins by importing necessary Python libraries:

```python
import gpxpy
import gpxpy.gpx
import numpy as np
import haversine as hs
import pandas as pd
import os
import gpxpy
import pandas as pd
from tqdm import tqdm
import json
```

These libraries are used for working with GPX files, numerical calculations, data manipulation, and progress tracking during processing.

2.1.2 Functions

The code defines several important functions:

gpx_to_csv

This function converts GPX data to CSV format and calculates various metrics.

```python
def gpx_to_csv(gpx_file_path, csv_file_path):
    with open(gpx_file_path, "r") as gpx_file:
        gpx = gpxpy.parse(gpx_file)

    route_info = []
    for track in gpx.tracks:
        for segment in track.segments:
            for point in segment.points:
                route_info.append({
                    "time": point.time,
                    "latitude": point.latitude,
                    "longitude": point.longitude,
                    "altitude": point.elevation
                })

    route_df = pd.DataFrame(route_info)

    route_df["altitude_diff"] = route_df["altitude"].diff()
    route_df["relative_elevation"] = route_df["altitude_diff"].cumsum()

    distances = [np.nan]
    speed = [np.nan]

    for i in range(1, len(route_df)):
        distances.append(haversine_distance(
            lat1=route_df.iloc[i - 1]["latitude"],
            lon1=route_df.iloc[i - 1]["longitude"],
            lat2=route_df.iloc[i]["latitude"],
            lon2=route_df.iloc[i]["longitude"]
        ))

        # #* speed
        time_diff = (route_df.iloc[i].time - route_df.iloc[i - 1].time).seconds
        distances_i = distances[i]

        # Handling division by zero
        if time_diff == 0:
            speed_i = 10  # Assign an appropriate default value
        else:
            speed_i = distances_i / time_diff

        speed.append(speed_i)

    route_df["distance"] = distances
    route_df["cum_distance"] = route_df["distance"].cumsum()/1e3
    route_df["speed"] = speed

    number_of_lifts = lift_checker(route_df)
    if number_of_lifts > 0:
        report.append({
            "file": csv_file_path[11:],
            "n": number_of_lifts,
            "sum_of_n": route_df["lift_path"].sum()/2
        })
        print("----------------------------------")
        print(f"The number of lifts detected on {csv_file_path[11:]} is {number_of_lifts}")
        print("----------------------------------")

    route_df = route_df.fillna(0)  # replace NANs with zero
    ######
    route_df.to_csv(csv_file_path, index=False)
    return route_df
```

It works as follows:

1. The function first parses the input GPX file using the *gpxpy* library and extracts key data like time, latitude, longitude and elevation into a Python dictionary for each point along the route.

2. It then converts this dictionary into a Pandas DataFrame to enable easier data manipulation.

3. Additional columns are created in the DataFrame to calculate elevation difference, cumulative elevation gain, distance between points, cumulative distance, and speed based on the time difference between points.

4. Potential divide-by-zero errors are handled when calculating speed.

5. A lift detection function is called to analyze the elevation profile and count the number of detected lifts along the route.

6. The number of detected lifts is tracked in a report.

7. Missing data in the DataFrame is filled with zeros.

8. Finally, the processed DataFrame is written out to a CSV file to save the updated route data.

9. The code returns the final DataFrame containing the enriched route data with statistics like speed, distance, elevation, and lift counts.

`haversine_distance`

In the previous function, the implementation leverages the functionality of two additional functions. Firstly, an auxiliary function is employed to compute the haversine distance, which quantifies the geographical distance between two distinct sets of latitude and longitude coordinates. This computation is facilitated through the utilization of the `haversine` library.

```python
def haversine_distance(lat1, lon1, lat2, lon2) -> float:
    distance = hs.haversine(
        point1=(lat1, lon1),
        point2=(lat2, lon2),
        unit=hs.Unit.METERS
    )
    return np.round(distance, 2)
```

lift_checker

Furthermore, an additional vital function comes into play. This function is dedicated to the identification of lift occurrences in the dataset. After loading the dataset of the lifts, two for loop is used. One iterates over the track and the other iterates over lifts. Then, lists are detected when the location of a track aligns with the location of a lift. In the event a lift is detected, this function augments the existing DataFrame of GPS data with two supplementary columns. The first column, designated as `lift?`, is equipped with boolean values (0 and 1) to indicate the presence of a lift at specific points. The second column, titled `lift_path`, serves as an indicator for lift pathways, allowing for the demarcation of paths corresponding to lift usage:

```python
def lift_checker(df):
    with open("lift_dataset.json", "r") as f:
        lifts = json.load(f)
    number_of_lifts = 0
    df["lift?"] = 0  # ? set the "lift?" column to zero
    df["lift_path"] = 0

    for i in range(len(df)):
        for lift in lifts:
            lift_location = lift["geoLocation"]["
                coordinatesLineString"]
            if df["latitude"][i] == lift_location[0] and df["
                longitude"][i] == lift_location[1]:
                number_of_lifts += 1
                df.loc[i, "lift?"] = 1
                df.loc[i-1:i, "lift_path"] = 1
    return number_of_lifts
```

convert_all_gpx_to_csv

This function processes all GPX files in a directory and converts them to CSV:

```python
def convert_all_gpx_to_csv(gpx_dir, csv_dir):
    gpx_files = [filename for filename in os.listdir(
        gpx_dir) if filename.endswith(".gpx")]
    progress_bar = tqdm(total=len(gpx_files), desc="Converting GPX files")
    for filename in gpx_files:
        gpx_file_path = os.path.join(gpx_dir, filename)
        csv_file_path = os.path.join(csv_dir, filename.replace(".gpx", ".csv"))
        gpx_to_csv(gpx_file_path, csv_file_path)
        progress_bar.update(1)
    progress_bar.close()
```

2.1.3 Main Execution

Eventually, the *convert_all_gpx_to_csv* function is invoked by providing it with the relevant directories for input GPX files (*gpx_dir*) and output CSV files (*csv_dir*). As the function iterates through each GPX file, it performs the necessary conversions and progress is visually indicated through status updates. After the conversion process concludes, a report is generated to catalog tracks that contain a minimum of one lift. This report is structured as a DataFrame and subsequently saved as a CSV file named *report.csv* within the designated data directory. The overall result is a seamless conversion of raw GPS data into a more structured and informative format, followed by the generation of a comprehensive report for further analysis.

```python
# Usage
gpx_dir = "./data/gpx_train/"
csv_dir = "./data/csv_train/"
convert_all_gpx_to_csv(gpx_dir, csv_dir)
print("Converting is finished.")

# Generating a report of the tracks that have at least one lift
report_df = pd.DataFrame(report)
report_df.to_csv("./data/report.csv", index=True)
```

2.2 Utilizing the CSV data

2.3 Conclusion

3 System

> *Innovation distinguishes between a leader and a follower.*
>
> — **Steve Jobs**
> (CEO Apple Inc.)

3.1 System Section 1

Fig. 3.1.: Figure example: *(a)* example part one, *(c)* example part two; *(c)* example part three

3.2 System Section 2

Fig. 3.2.: Another Figure example: *(a)* example part one, *(c)* example part two; *(c)* example part three

3.3 System Section 3

3.4 Conclusion

4 Results

4.1 Concepts Section 1

4.2 Concepts Section 2 with a very very long title that illustrates how long section titles are handled in the footer

4.3 Concepts Section 3

4.4 Conclusion

Conclusion 5

5.1 System Section 1

5.2 System Section 2

5.3 Future Work

Bibliography

[1] Ethem Alpaydin. *Introduction to Machine Learning*. 3rd ed. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2014 (cit. on p. 3).

[2] Michael W Browne. "Cross-Validation Methods". In: *Journal of Mathematical Psychology* 44.1 (2000), pp. 108–132 (cit. on p. 5).

[3] JA Cook and J Ranstam. "Overfitting". In: *Journal of British Surgery* 103.13 (2016), pp. 1814–1814 (cit. on p. 4).

[4] David Hand, Heikki Mannila, and Padhraic Smyth. "Principles of data mining. 2001". In: *MIT Press. Sections* 6.3 (2001), pp. 2–6 (cit. on p. 2).

[5] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009 (cit. on pp. 3, 4).

[6] Douglas M. Hawkins. "The Problem of Overfitting". In: *Journal of Chemical Information and Computer Sciences* 44.1 (2004), pp. 1–12 (cit. on p. 4).

[7] Michael I Jordan and Tom M Mitchell. "Machine learning: Trends, perspectives, and prospects". In: *Science* 349.6245 (2015), pp. 255–260 (cit. on p. 3).

[8] V. Roshan Joseph and Akhil Vakayil. "SPlit: An Optimal Method for Data Splitting". In: *Technometrics* 64.2 (2022), pp. 166–176 (cit. on p. 4).

[9] Tom M Mitchell. *Machine learning*. 1997 (cit. on p. 2).

[10] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012 (cit. on p. 3).

[11] Lutz Prechelt. "Early Stopping — But When?" In: *Neural Networks: Tricks of the Trade: Second Edition*. Ed. by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 53–67 (cit. on p. 5).

[12] Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007 (cit. on p. 5).

[13] Xue Ying. "An Overview of Overfitting and its Solutions". In: *Journal of Physics: Conference Series* 1168.2 (Feb. 2019), p. 022022 (cit. on p. 4).

[14] Zhi-Hua Zhou. *Machine learning*. Springer Nature, 2021 (cit. on pp. 2, 3).

List of Figures

List of Tables

List of Listings

Example Appendix

A

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.1 Appendix Section 1

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Alpha	Beta	Gamma
0	1	2
3	4	5

Tab. A.1.: This is a caption text.

A.2 Appendix Section 2

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Alpha	Beta	Gamma
0	1	2
3	4	5

Tab. A.2.: This is a caption text.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Colophon

This thesis was typeset with LaTeX 2$_{\varepsilon}$. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at http://cleanthesis.der-ric.de/.

Declaration

You can put your declaration here, to declare that you have completed your work solely and only with the help of the references you mentioned.

Padova, September 05, 2023

Nima Karimi